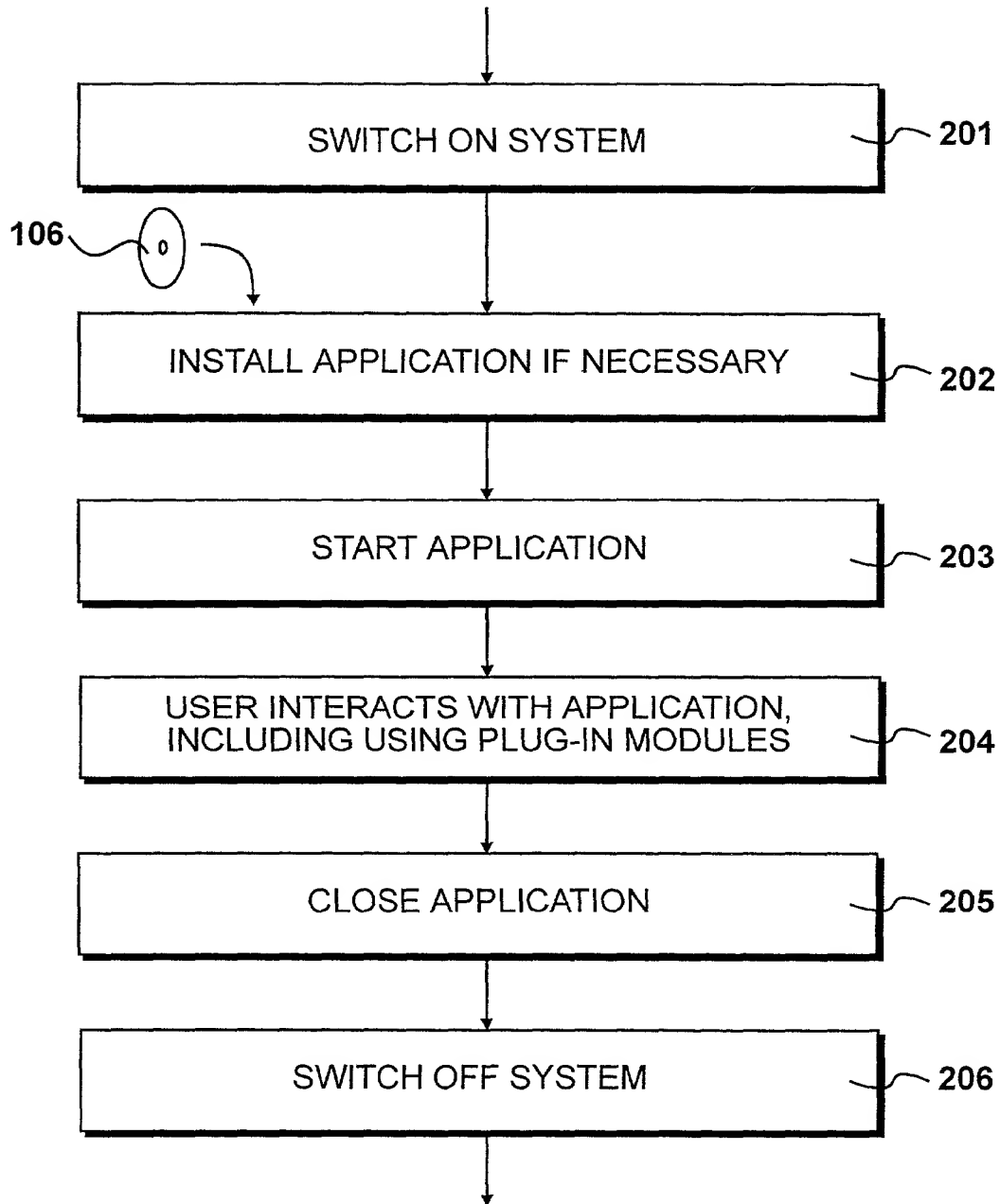


Figure 1

102 105 104 101 106 103

2/20

*Figure 2*

3/20

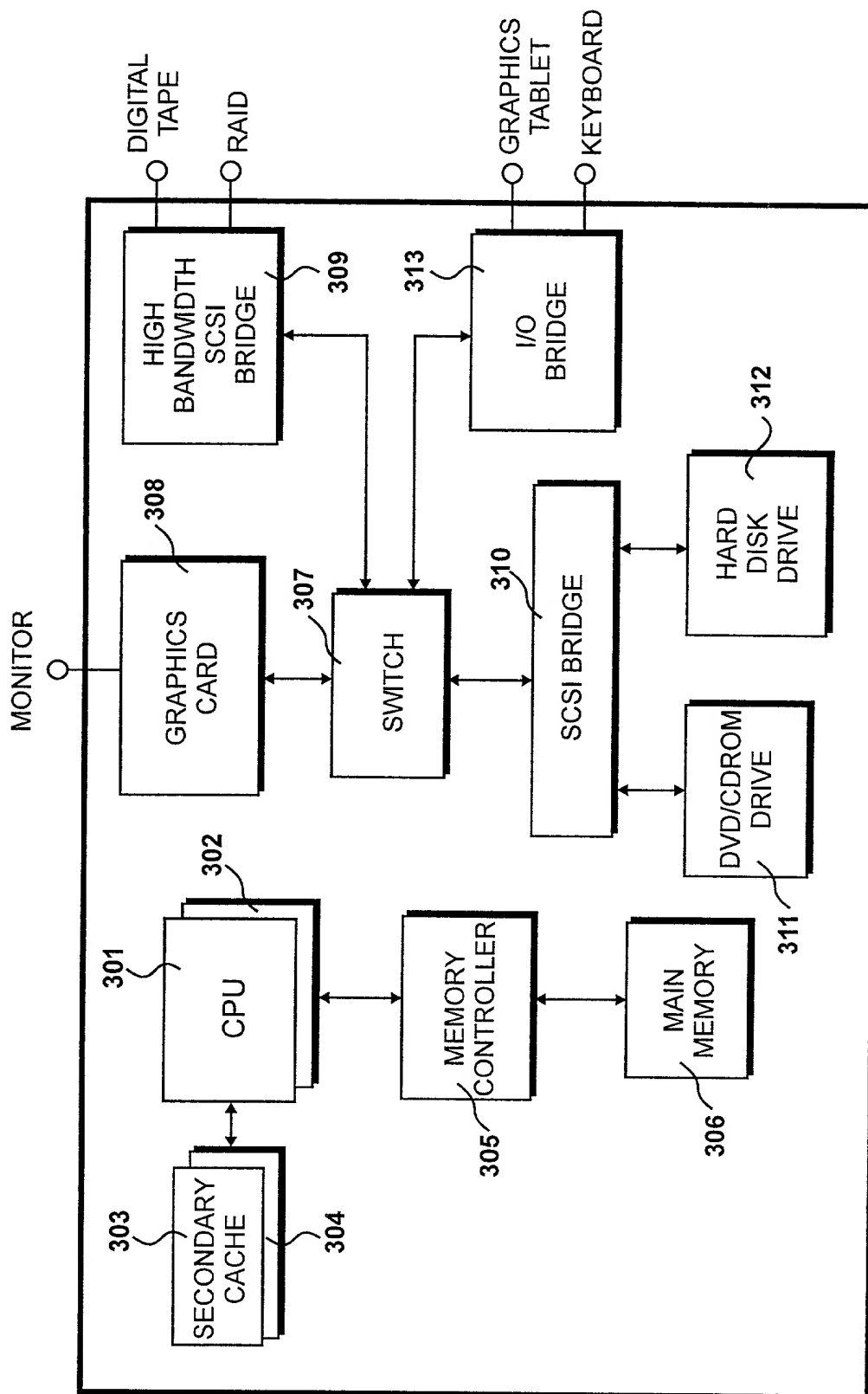
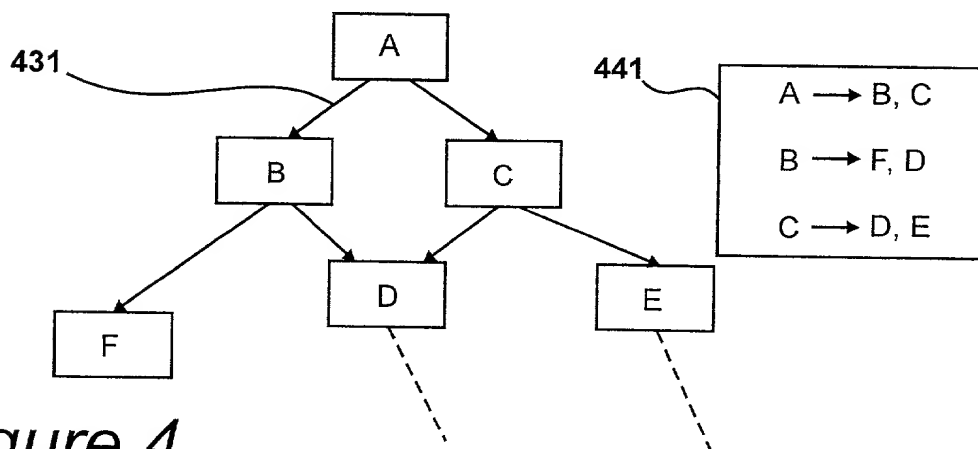
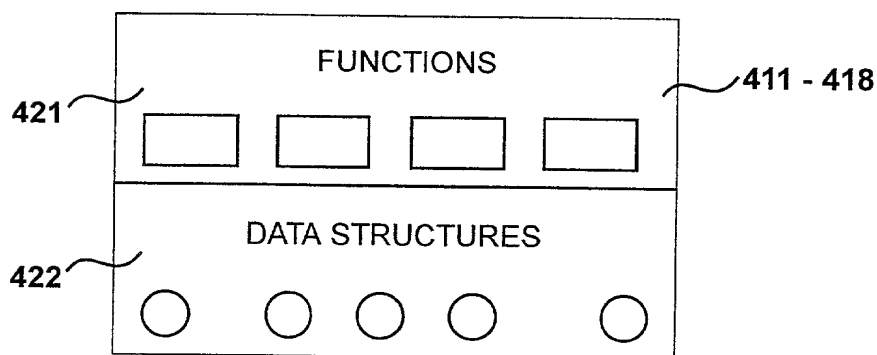
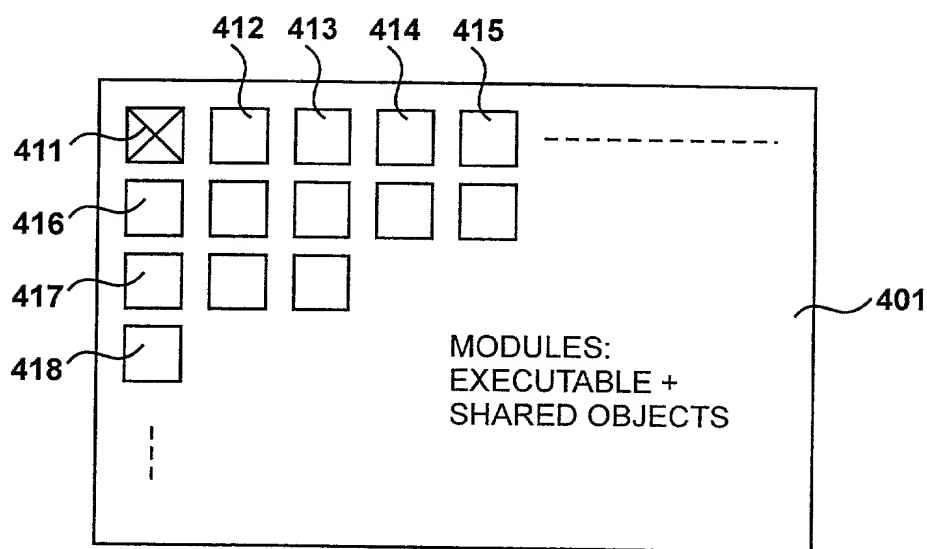


Figure 3

4/20

*Figure 4*

5/20

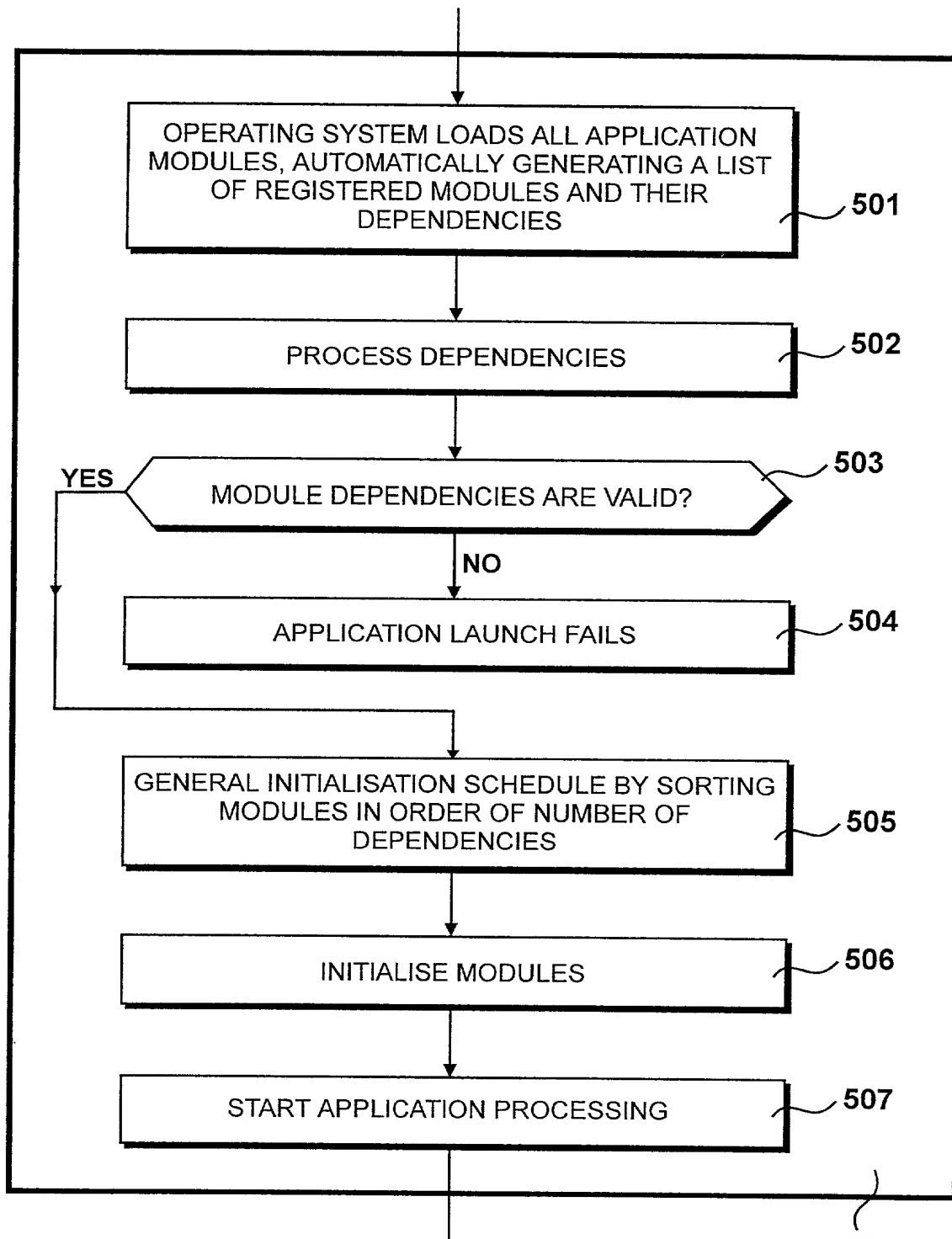


Figure 5

203

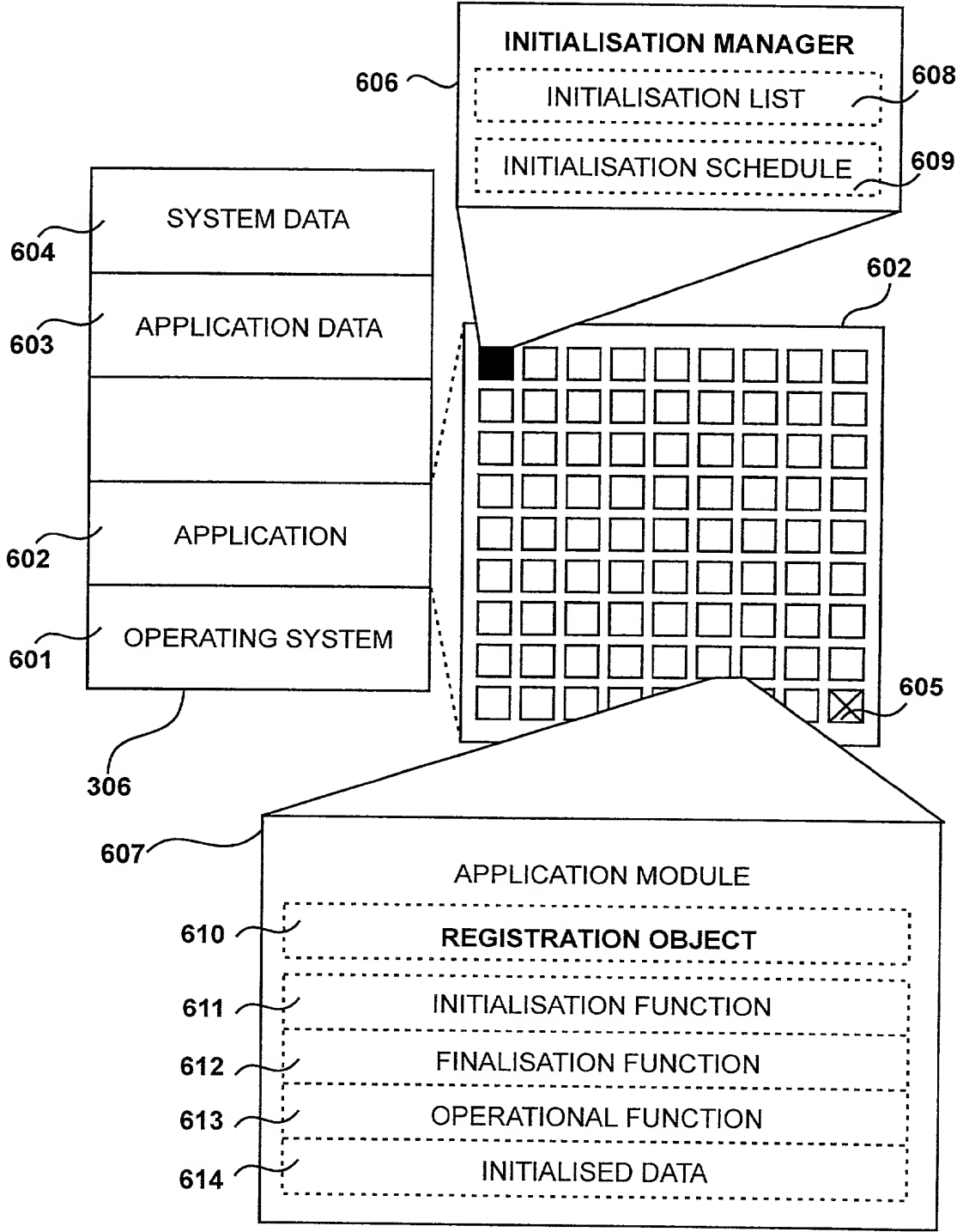
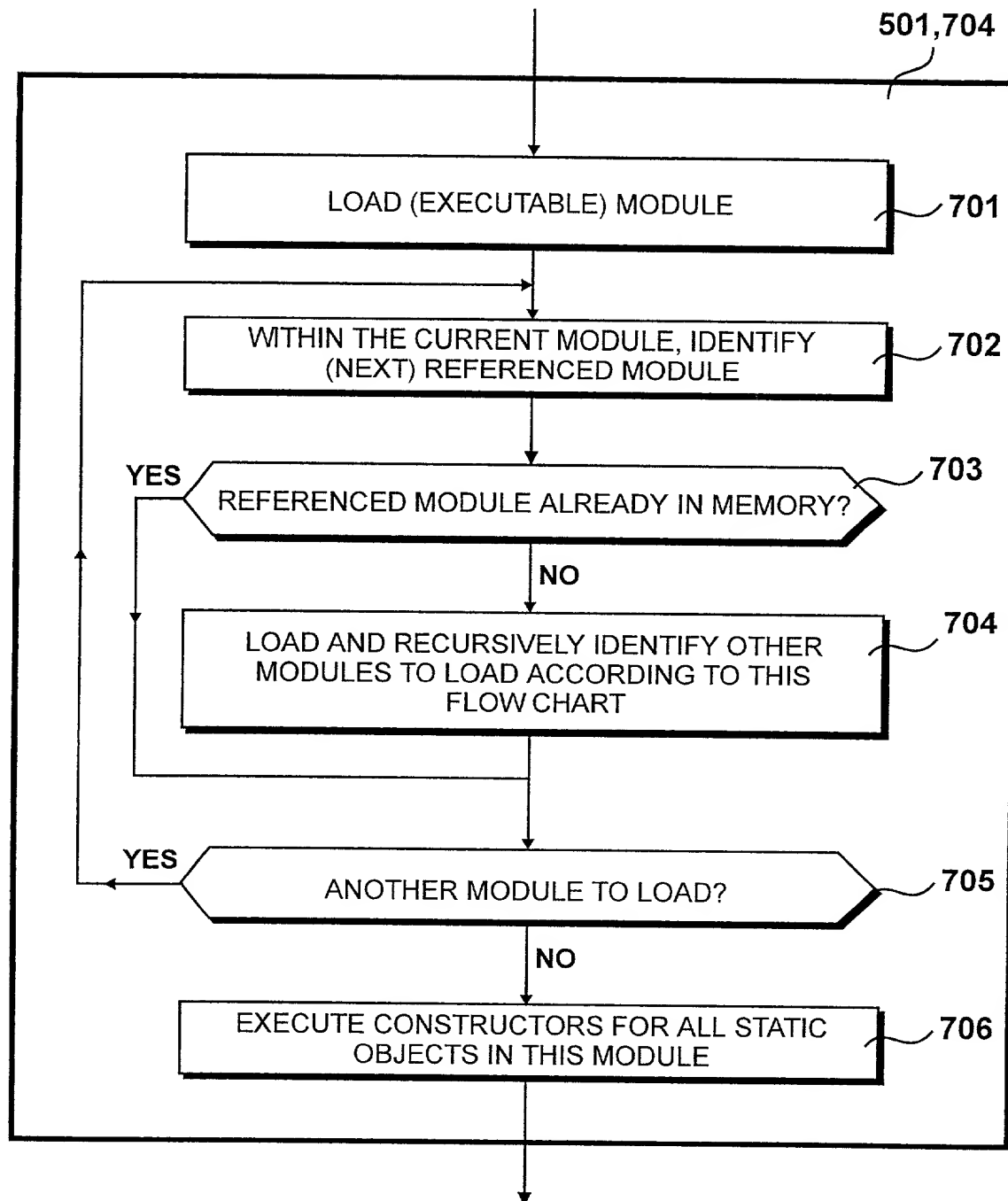


Figure 6

7/20

*Figure 7*

8/20

```
// Filename: foo.cpp

#include <database.h>
#include <osal.h>
#include <initialize.h>

#include "foo.h"

// Static objects 801
InInit<foo> _instance; //registration object

// Constructor for registration object
InInit<foo>::InInit () { 802

// Registering Dependencies
addDependency(InInit<database>::getType());actual call pattern;
registerDependency(typeInfo(InInit<osal>).typeID());
registerDependency(typeInfo(InInit<initialize>).typeID());

}

Module initialisation and finalisation functions
InInit<foo>::performInitialise()
    () { 803

// whatever initialisation this module requires

}

void InInit<foo>::performFinalize()
    () { 804

// whatever finalisation this module requires

}

// All other module functions from this point on ...
```

Figure 8

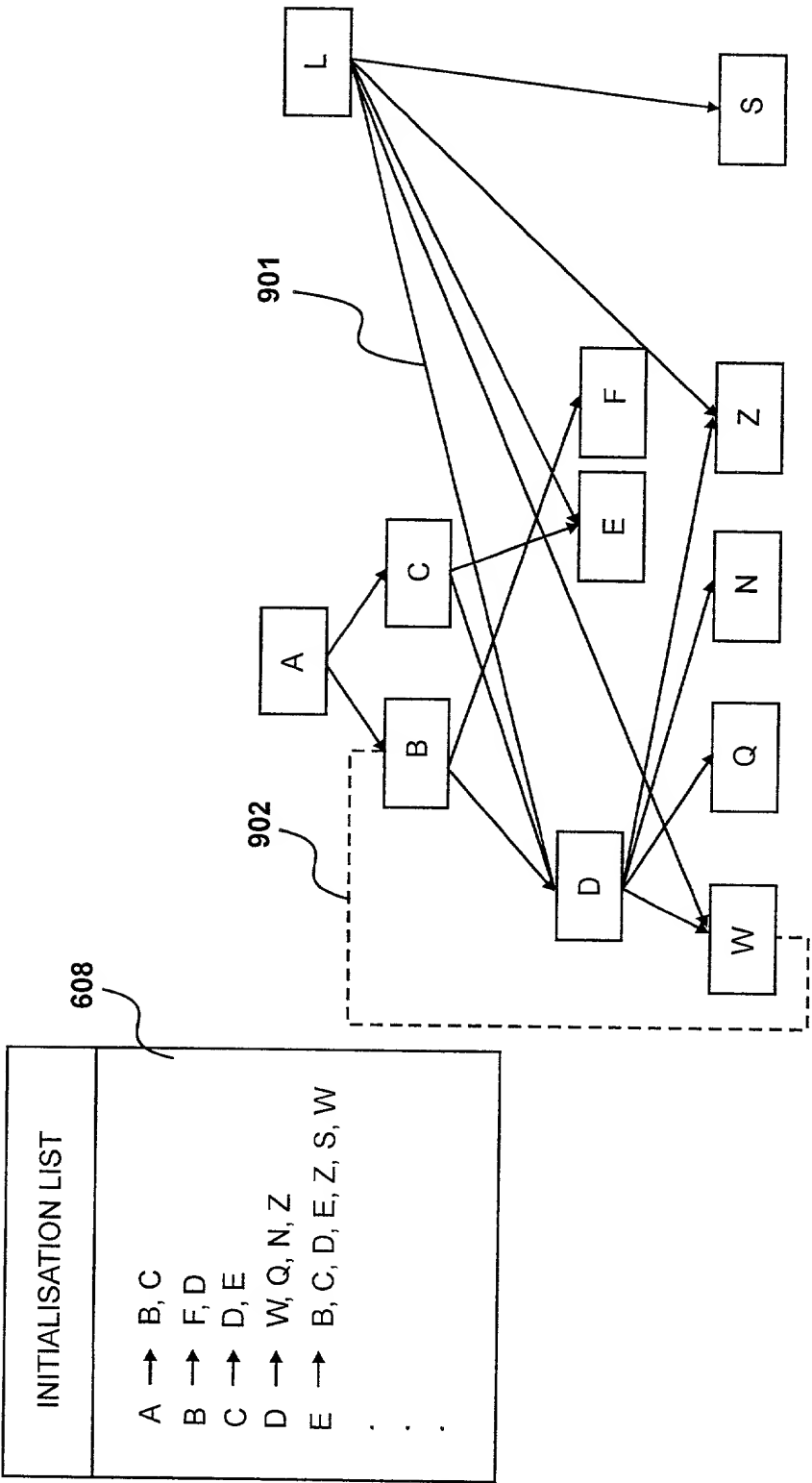
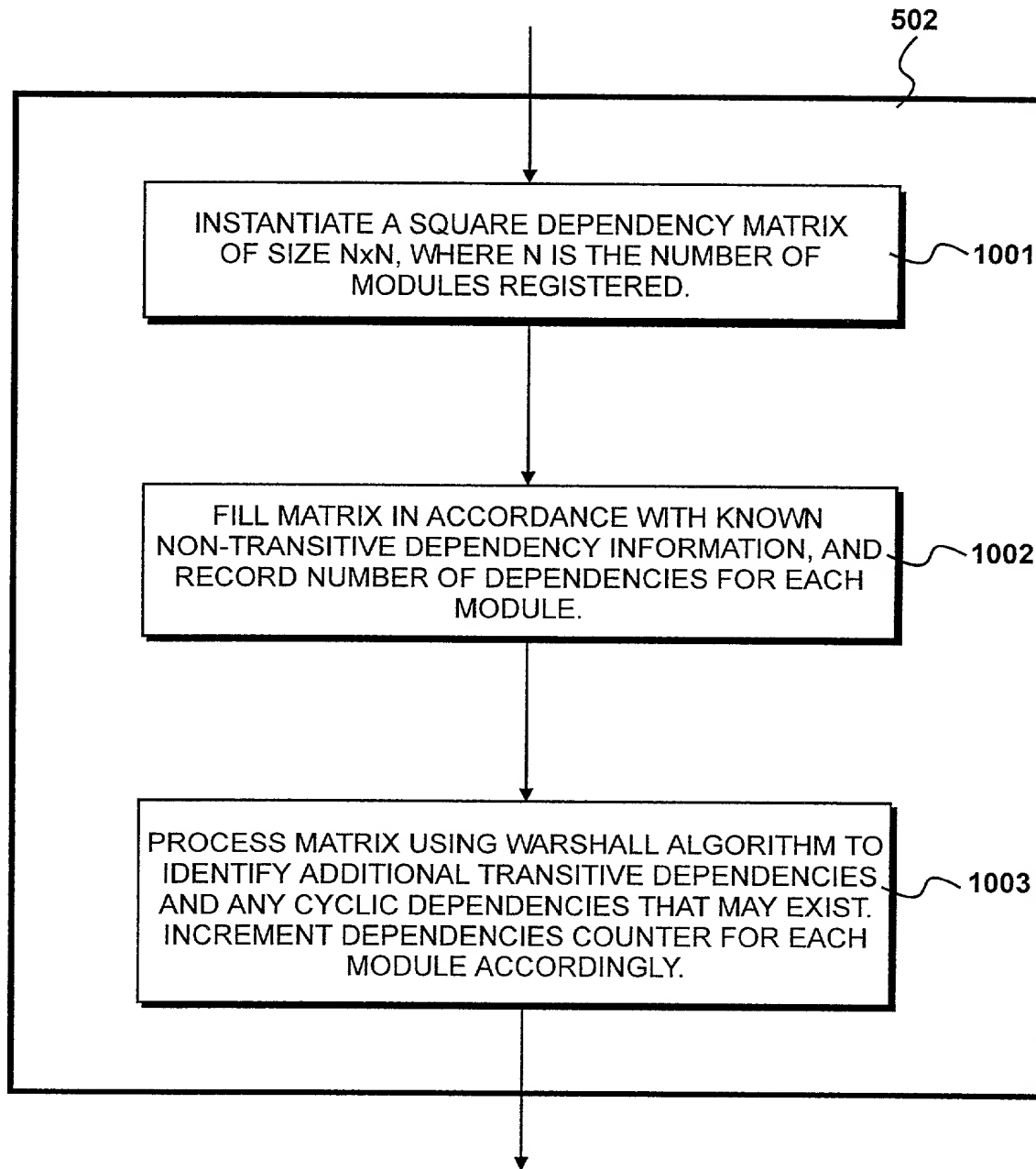


Figure 9

10/20

*Figure 10*

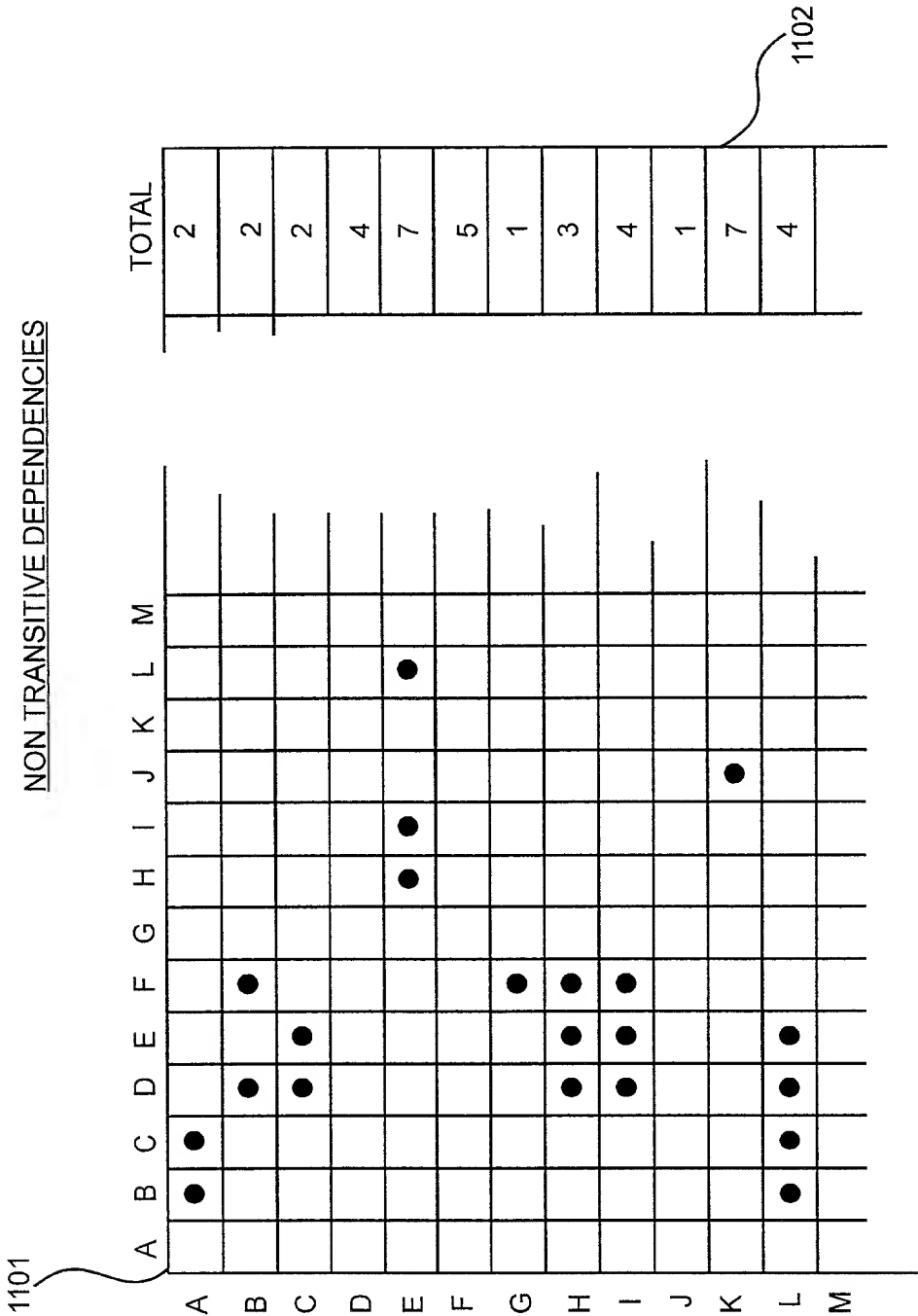


Figure 11

12/20

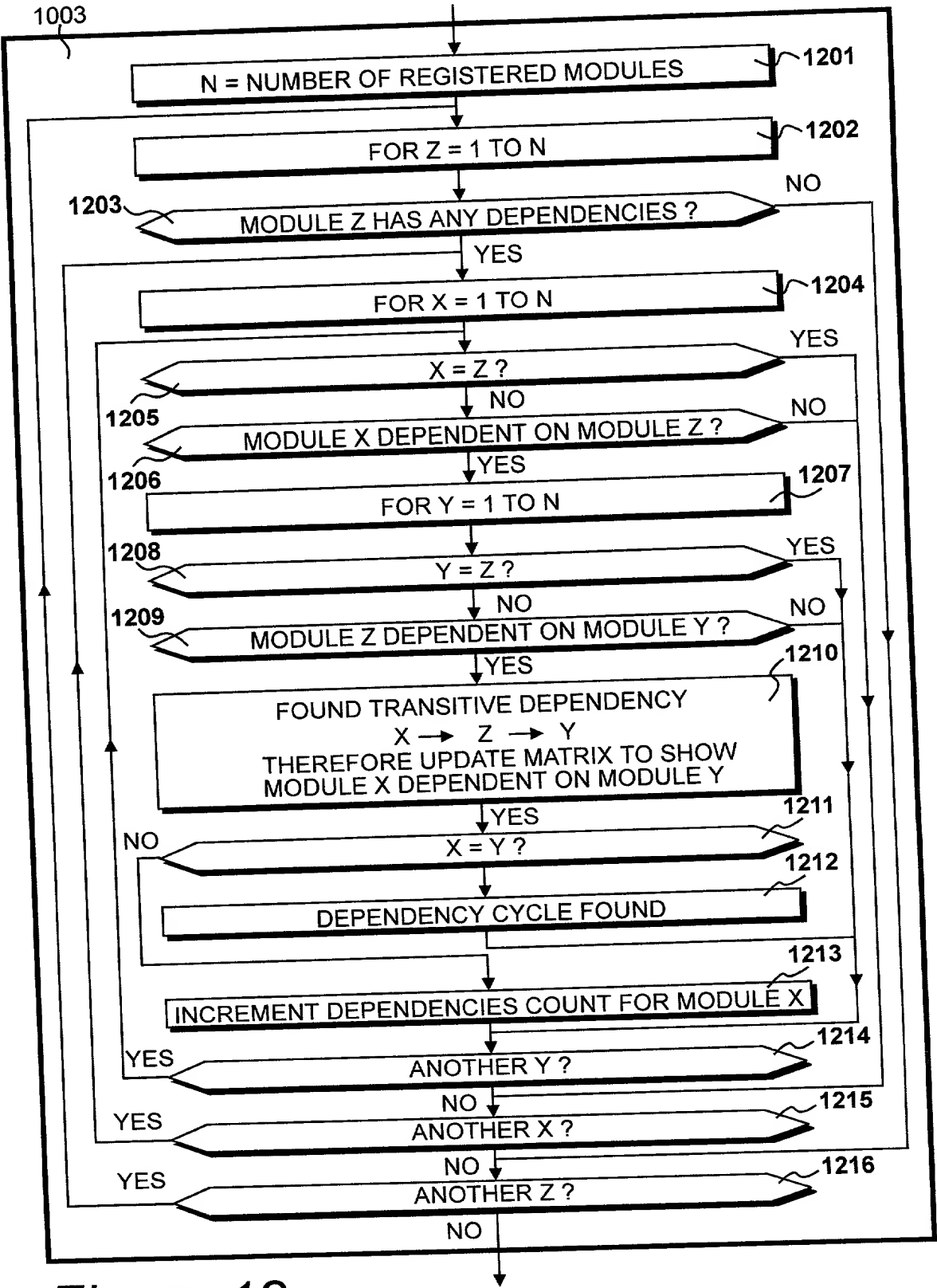


Figure 12

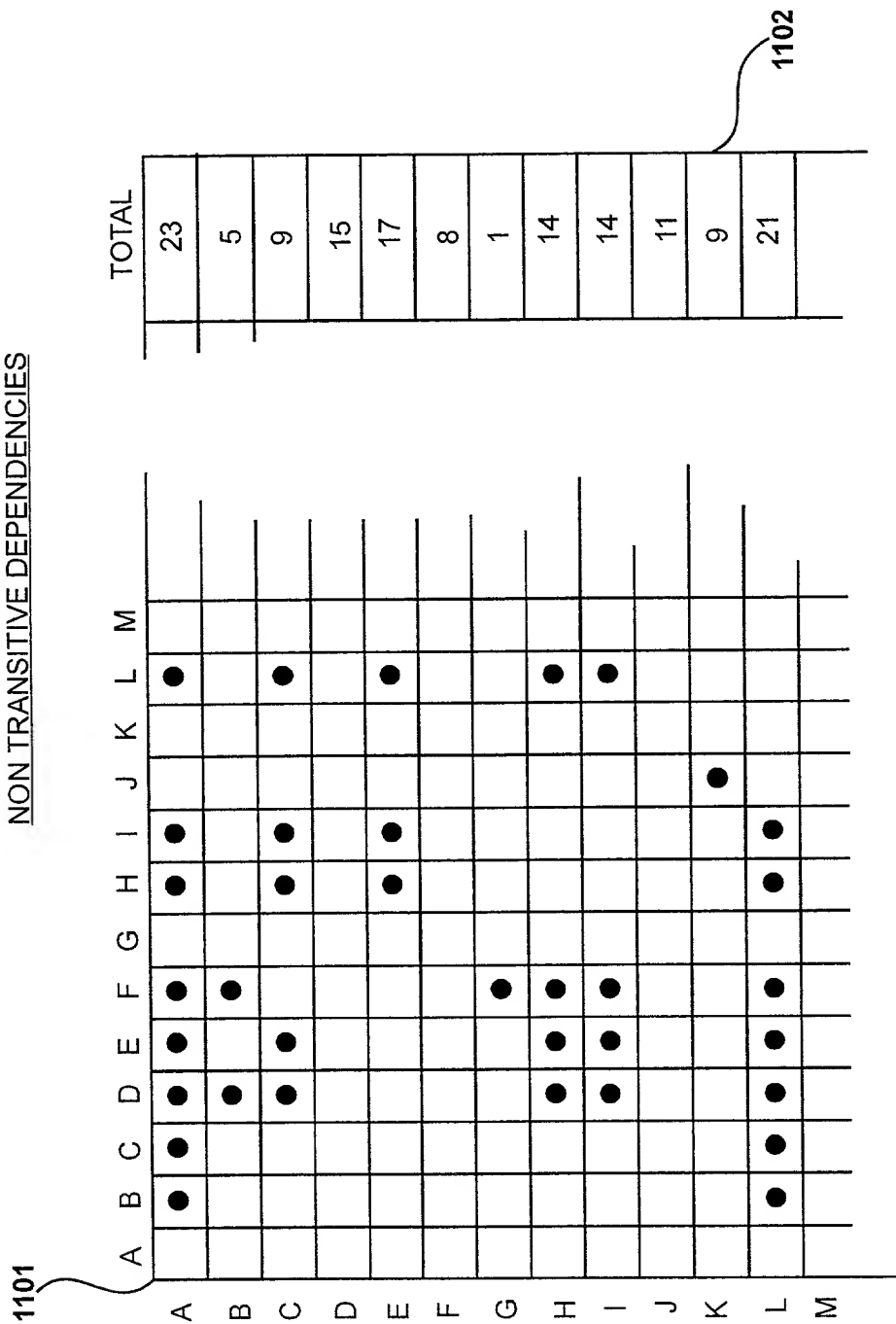


Figure 13

Continued

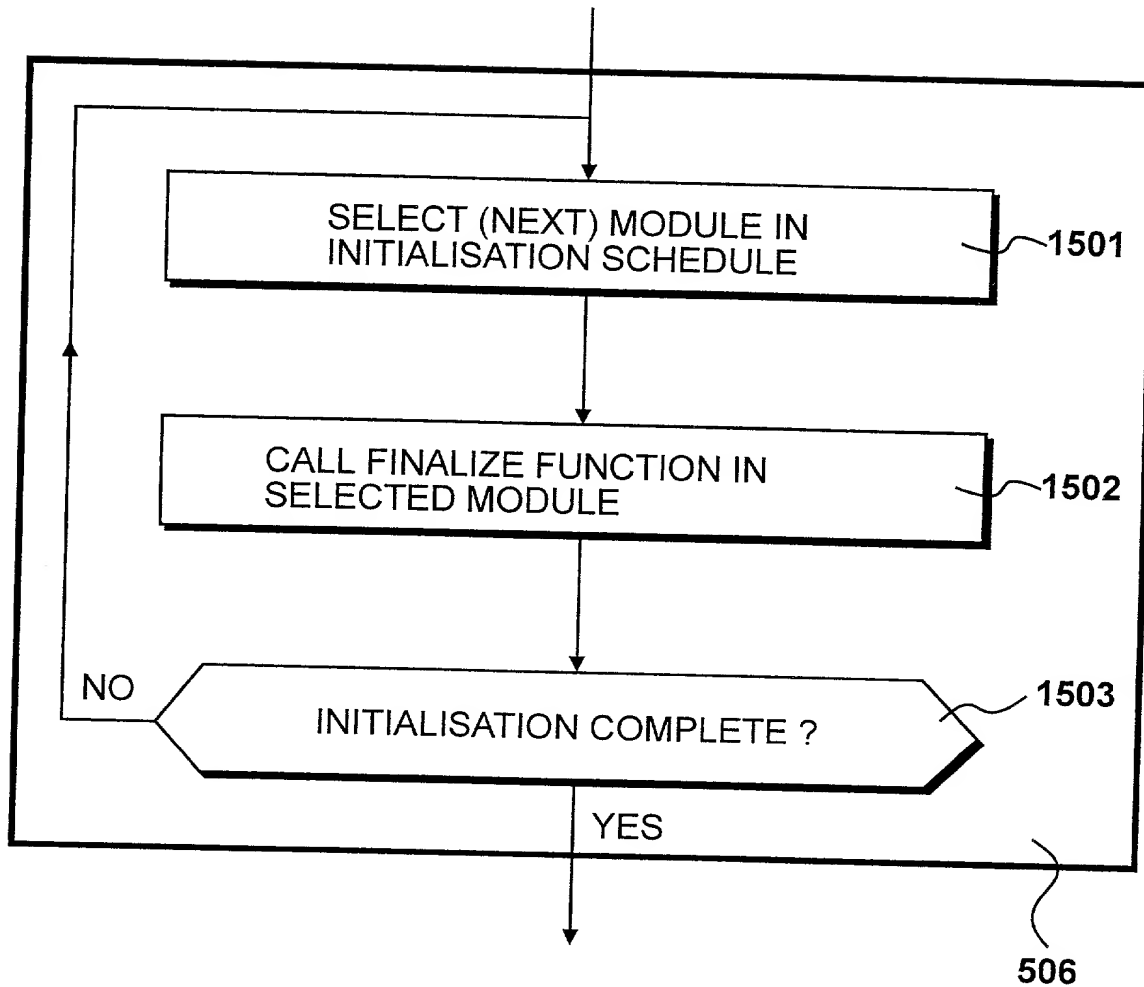
INITIALISATION SCHEDULE

MODULE	DEPENDENCIES	RANK
G	1	1
B	5	2
F	8	3
C	9	4
K	9	5
J	11	6
H	14	7
I	14	8
D	15	9
E	17	10
L	21	11
A	23	12

609

Figure 14

15/20

*Figure 15*

16/20

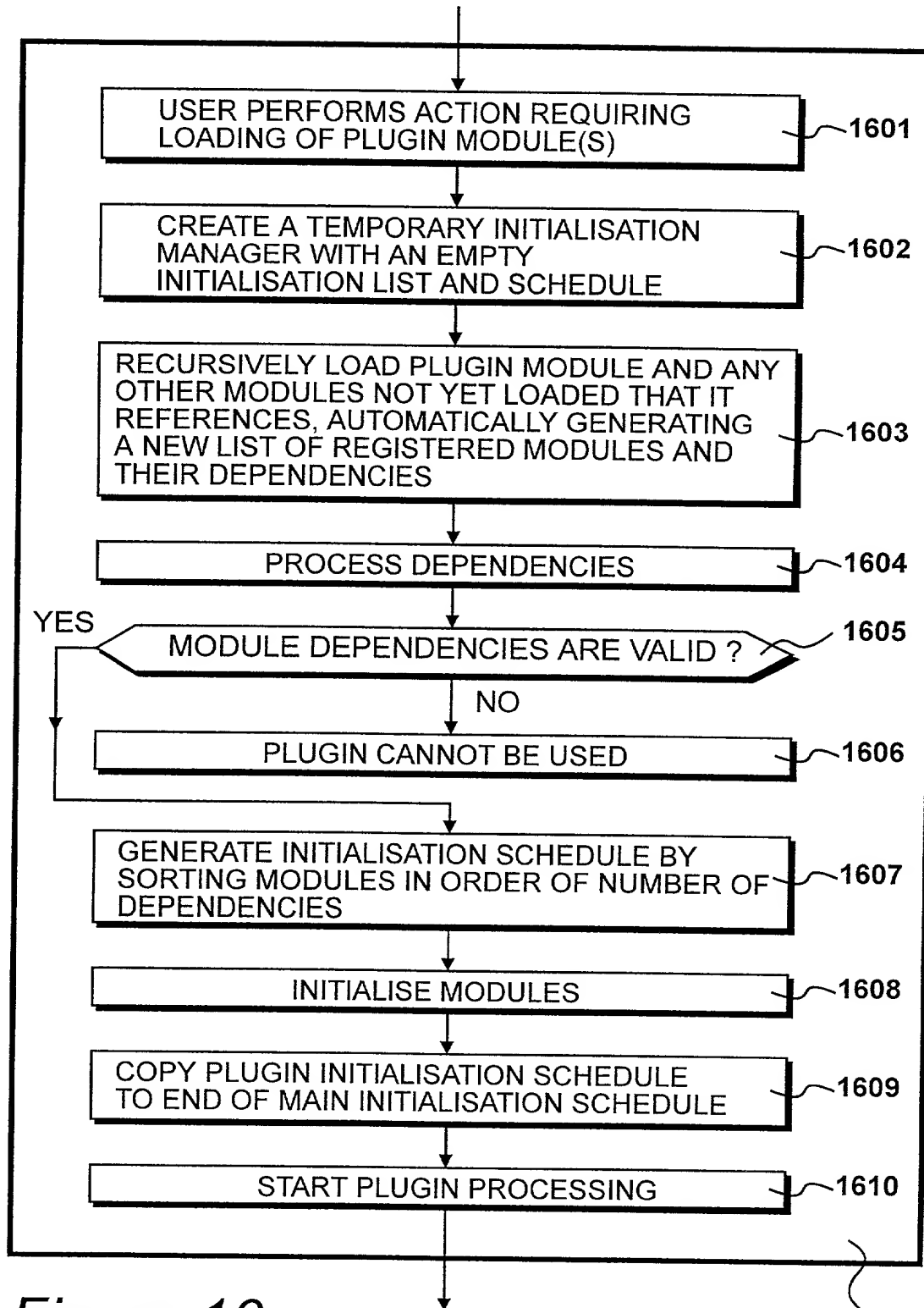


Figure 16

204

FOUO 20200600

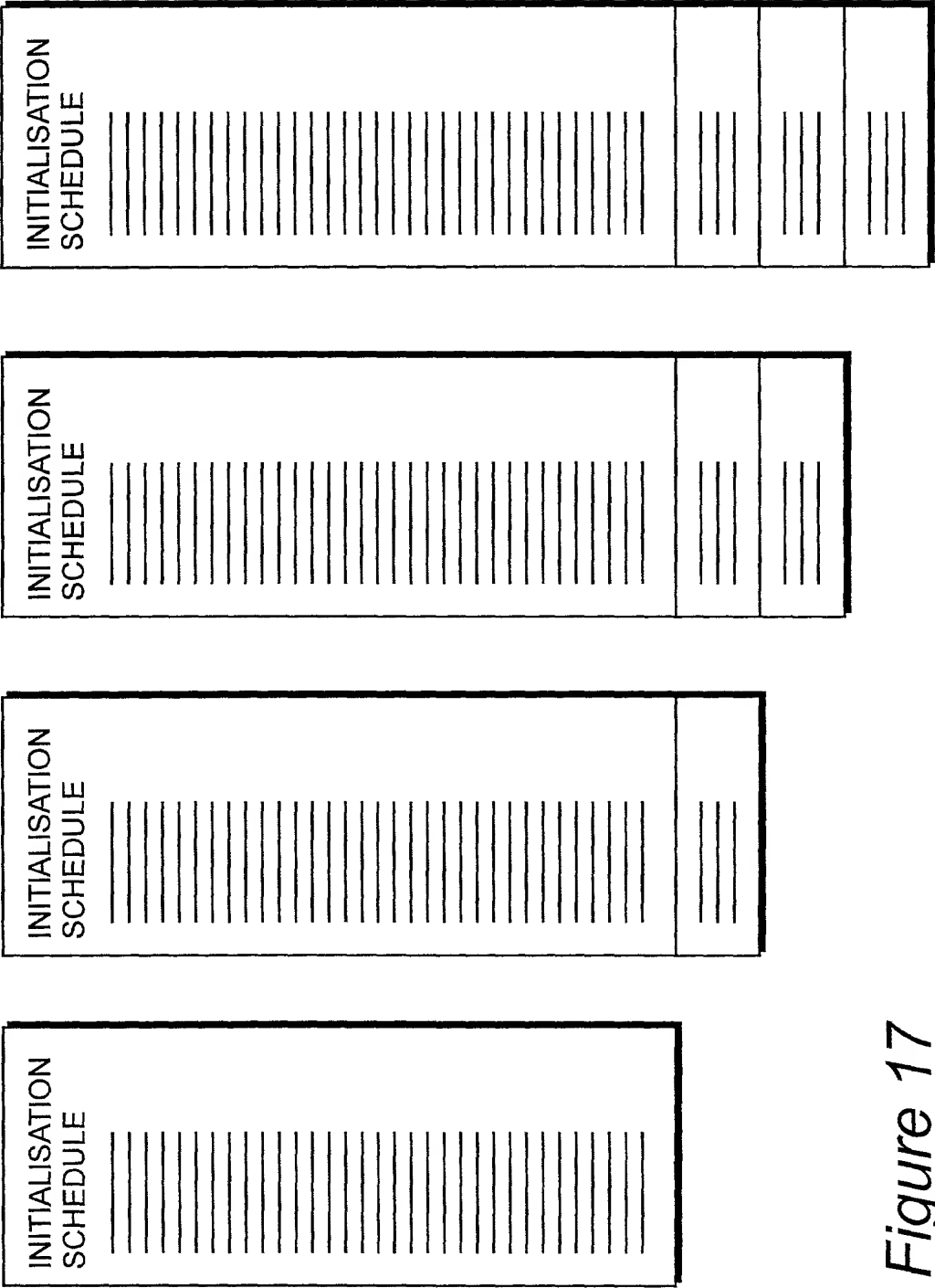
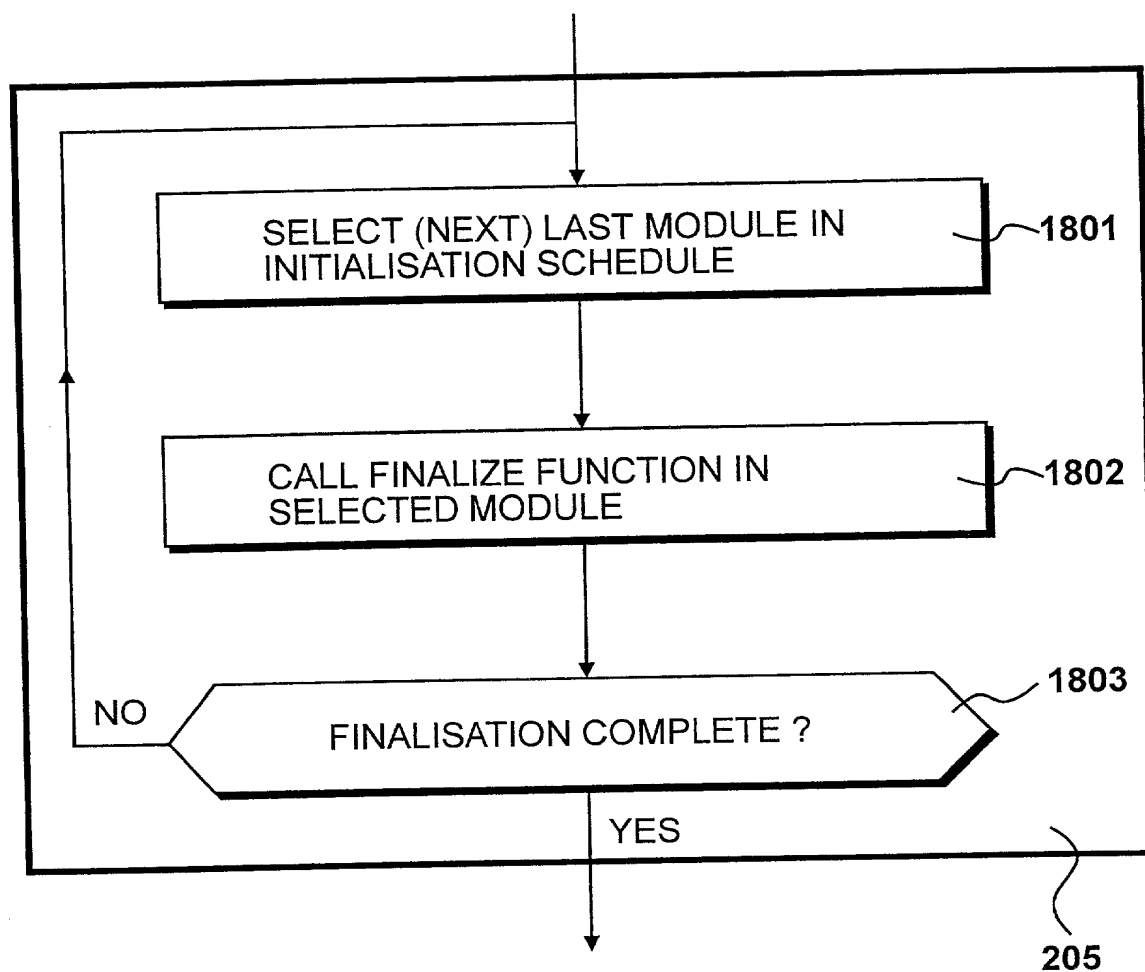


Figure 17

18/20

*Figure 18*

19/20

```

// Filename: main.cpp

#include <fastmath.h>
#include <osal.h>
#include <initialize.h>

#include "main.h"

// Static objects

InInit<main> _instance; //registration object

// Constructor for registration object

InInit<main>::InInit () {

    registerDependency(typeInfo(InInit<fastmath>).typeID());
    registerDependency(typeInfo(InInit<osal>).typeID());
    registerDependency(typeInfo(InInit<initialize>).typeID());

}

main() {
    InitGuard ig;

    // the rest of the main function goes here

    // Other functions, include initialise() and finalise()
    // go here ...

```

Figure 19

20/20

```
InitGuard :: InitGuard () {  
  
    InInitManager.initialize () ;  
  
}
```

Figure 20

```
void loadPlugin () {  
    ~ 2101  
    ReInitGuard rig;  
    dlopen ("Plugin1");  
    dlopen ("Plugin2");  
    ~ 2102  
}
```

Figure 21

```
ReInitGuard :: ReInitGuard () {  
  
    ReInitManager.initialize () ;  
  
}
```

Figure 22